Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Лабораторная работа №3

по дисциплине «Базовые информационные технологии»

Тема: «Объекты и прототипы в JavaScript. ООП. Обработка исключений в JavaScript.»

Выполнил:

студент гр. ПРИ-123

А.Ц. Нямаа

Принял:

Ст. пр. кафедры ИСПИ

О.Н. Шамышев

Владимир, 2025 г.

ЦЕЛЬ РАБОТЫ:

Научиться решать задачи с помощью объектов и создавать конструкторы объектов с набором общих методом. Решить задачи на ООП, написать код с использованием классов. Выполнить задачи на обработку исключений.

ЗАДАНИЕ 1 (вариант 12).

Создать конструктор объекта с определенным набором свойств и методов. Создать несколько объектов и вывести их на сайт.

ВЫПОЛНЕНИЕ РАБОТЫ:

Листинг index.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Overpass:ital,wght@0,100..900;1,100..900&display=swap"

        rel="stylesheet">

    <title>Clock Shop</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <header>

        <div class="leftText">

            <h1> ДИЗАЙНЕРСКИЙ ЧАСЫ </h1>

            <p> ПОЛНОСТЬЮ РУЧНОЕ ИЗГОТОВЛЕНИЕ </p>

            <button class="buyHeader buttonToChange openModel" data-price="900000"> КУПИТЬ </button>

            <dialog class="dialogMain">

                <form class="orderForm">

                    <h2>Оформление заказа</h2>

                    <p class="hidden" id="validSuccess"></p>

                    <p class="hidden" id="validError"></p>

                    <p><label for="name">Имя:</label>

                        <input type="text" id="name" name="name" placeholder="Имя" required>

                    <p class="hidden text-danger" id="nameP"></p>

                    </p>

                    <p><label for="secondName">Фамилия:</label>

                        <input type="text" id="secondName" name="secondName" placeholder="Фамилия" required>

                    <p class="hidden text-danger" id="secondNameP"></p>

                    </p>

                    <p><label for="thirdName">Отчество:</label>

                        <input type="text" id="thirdName" name="thirdName" placeholder="Отчество">

                    <p class="hidden text-danger" id="thirdNameP"></p>

                    </p>

                    <p><label for="address">Адрес доставки:</label>

                        <input type="text" id="address" name="address" placeholder="Ул. Пушкино, д. 2, кв. 1" required>

                    </p>

                    <p><label for="quantity">Количество:</label>

                        <input type="number" id="quantity" name="quantity" placeholder="1" min="1" max="10" required>

                    </p>

                    <p><label for="promo">Промокод:</label>

                        <input type="text" id="promo" placeholder="\*Промокод\*" name="promo">

                    </p>

                    <p class="priceHeader"></p>

                    <input type="submit" value="Заказать!">

                </form>

                <button id="closeModel" class="closeModel">&#x2715 </button>

            </dialog>

        </div>

        <div class="rightText">

            <p> JACOB & CO <br>EPIC X TOURBILON BAGUETTE <br>900.000$</p>

            <img src="imgs/mainPhotoWatches.png">

        </div>

    </header>

    <main>

        <div class="textInMain">

            <p>ХИТЫ</p>

            <p>ПОСЛЕДНИЕ ПОСТУПЛЕНИЯ</p>

        </div>

        <div class="productsList"> </div>

        <div class="addModule">

            <div class="leftSide">

                <div class="textInMain addText">

                    <p>РАБОТА С ОБЪЕКТАМИ</p>

                </div>

                <form class="decorateBlock" id="objectId">

                    <p><label for="idObjectValue">Id объекта</label>

                        <input type="number" required for="idObjectValue" placeholder="1.." id="idObjectValueInput">

                    </p>

                </form>

                <form class="decorateBlock" id="objectFuncs">

                    <p><label for="priceObjectValue">Цена</label>

                        <input type="number" required for="priceObjectValue" placeholder="100$.."

                            id="priceObjectValueInput">

                    </p>

                    <p><label for="nameObjectValue">Название</label>

                        <input type="text" required for="nameObjectValue" placeholder="Название..."

                            id="nameObjectValueInput">

                    </p>

                </form>

                <form class="decorateBlock" id="objcetButtons">

                    <button id="PriceChanger"> Изменить цену! </button>

                    <button id="NameChanger"> Изменить name! </button>

                </form>

                <p class="objectWorkResult"></p>

            </div>

        </div>

    </main>

    <script src="script.js"></script>

</body>

</html>

Листинг script.js:

// Смена цвета кнопки при наведении

const button = document.body.querySelectorAll(".buttonToChange");

for (let i = 0; i < button.length; i++) {

    button[i].addEventListener("mouseover", function () {

        button[i].classList.add("changebackground");

    });

    button[i].addEventListener("mouseout", function () {

        button[i].classList.remove("changebackground");

    });

}

// Модальное окно

const dialog = document.querySelector(".dialogMain");

const html = document.querySelector("html");

const priceHeader = document.querySelector(".priceHeader");

document.querySelector("#closeModel").onclick = function () {

    dialog.close();

    html.classList.remove("changehtml");

};

//Изменения цены

const formElement = document.querySelector(".orderForm");

const quantity = document.querySelector("#quantity");

const promo = document.querySelector("#promo");

const promoList = {

    "10procentSaleWOW": 0.9,

    "15yearShop": 0.85,

};

function updatePrice() {

    let quantityVal = 1;

    const price = null;

    if (quantity.value > 1) {

        quantityVal = quantity.value;

    }

    let discount = promo.value in promoList ? promoList[promo.value] : 1;

    priceHeader.textContent = `Цена: ${createPrice(price, quantityVal, discount).toLocaleString()}$ (${quantityVal} шт, скидка ${Math.round((1 - discount) \* 100)}%)`;

}

function createPrice(priceFunc, quantityFunc, discountFunc) {

    let finalPrice = priceFunc \* quantityFunc \* discountFunc;

    return finalPrice;

}

quantity.addEventListener("input", updatePrice);

promo.addEventListener("input", updatePrice);

// Отправка формы

formElement.addEventListener("submit", (event) => {

    event.preventDefault();

    const formData = new FormData(formElement);

    const name = formData.get("name");

    const surname = formData.get("secondName");

    const thirdName = formData.get("thirdName");

    const address = formData.get("address");

    if (!checker(name, "nameP") || !checker(surname, "secondNameP") || !checker(thirdName, "thirdNameP")) {

        document.getElementById("validError").textContent = "Ошибка! Некоторый данный были введены неправильно";

        document.getElementById("validError").classList.remove("hidden");

        document.getElementById("validSuccess").classList.add("hidden");

    } else {

        document.getElementById("validSuccess").textContent = `Спасибо, ${name} ${surname} ${thirdName}!\nВаш заказ на адрес - ${address} в количестве ${quantity.value}шт будет исполнен в ближайшее время!`;

        document.getElementById("validSuccess").classList.remove("hidden");

        document.getElementById("validError").classList.add("hidden");

    }

});

function checker(str, idEl) {

    const regexStr = /^[A-Za-zА-Яа-яЁё]+$/;

    if (!regexStr.test(str)) {

        document.getElementById(idEl).textContent = "Не должно быть цифр!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    if (str.length > 50 || str.length < 3) {

        document.getElementById(idEl).textContent = "От 3 до 50 символов!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    document.getElementById(idEl).textContent = "";

    return true;

}

//Объекты

let productList = [];

function Product(id, name, price, imgPath) {

    this.id = id;

    this.name = name;

    this.price = price;

    this.imgPath = imgPath;

}

productList.push(

    new Product(1, "Apple watch ulta 2", 700, "imgs/appleulta.png"),

    new Product(2, "Rolex Day-Date 36", 117250, "imgs/rolexDayDate36.png"),

    new Product(3, "King Seiko Automatic", 2000, "imgs/KingSeiko.png"),

    new Product(4, "Paket Philippe 5270/1R", 230000, "imgs/PatekPhilippe.png"),

    new Product(5, "Rolex DateJust 31", 51900, "imgs/rolexDateJust31.png"),

    new Product(6, "Patek Philippe 6300/400G", 9000000, "imgs/PatekPhilippeBlackWhite.png"),

    new Product(7, "Apple watch series 10", 450, "imgs/appleseries10.png"),

    new Product(8, "Jacob&Co Palatial Classic", 16500, "imgs/Jacob&Co.png")

);

//Работа с объектами

Product.prototype.changePrice = function(newPrice){

    this.price = newPrice;

}

Product.prototype.changeName = function(newName){

    this.name = newName;

}

document.getElementById("PriceChanger").addEventListener("click", function(event){

    event.preventDefault();

    const tempId = parseInt(document.getElementById("idObjectValueInput").value);

    const tempPrice = parseFloat(document.getElementById("priceObjectValueInput").value);

    productList[tempId].changePrice(tempPrice);

    const productUnits = document.querySelectorAll(".productUnit");

    productUnits[tempId].querySelector("p:nth-child(3)").textContent = `${productList[tempId].price}$`;

});

document.getElementById("NameChanger").addEventListener("click", function(event){

    event.preventDefault();

    const tempId = parseInt(document.getElementById("idObjectValueInput").value);

    const tempName = document.getElementById("nameObjectValueInput").value;

    productList[tempId].changeName(tempName);

    const productUnits = document.querySelectorAll(".productUnit");

    productUnits[tempId].querySelector("p:nth-child(2)").textContent = `${productList[tempId].name}`;

});

//Генерация html

const productListContainer = document.querySelector(".productsList");

productList.forEach((product) => {

    const productUnit = document.createElement("div");

    productUnit.classList.add("productUnit");

    const img = document.createElement("img");

    img.src = product.imgPath;

    const name = document.createElement("p");

    name.textContent = product.name;

    const price = document.createElement("p");

    price.textContent = `${product.price}$`;

    const buyButton = document.createElement("button");

    buyButton.classList.add("buttonUnit", "buttonToChange", "openModel");

    buyButton.textContent = "Купить!";

    buyButton.dataset.price = product.price;

    buyButton.addEventListener("click", function () {

        dialog.showModal();

        html.classList.add("changehtml");

        priceHeader.textContent = `Цена: ${product.price.toLocaleString()}$`;

    });

    productUnit.appendChild(img);

    productUnit.appendChild(name);

    productUnit.appendChild(price);

    productUnit.appendChild(buyButton);

    productListContainer.appendChild(productUnit);

});

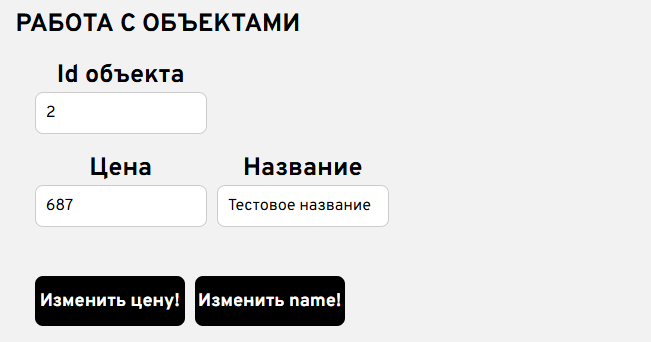


Рисунок 1. Блок работы с объектами.

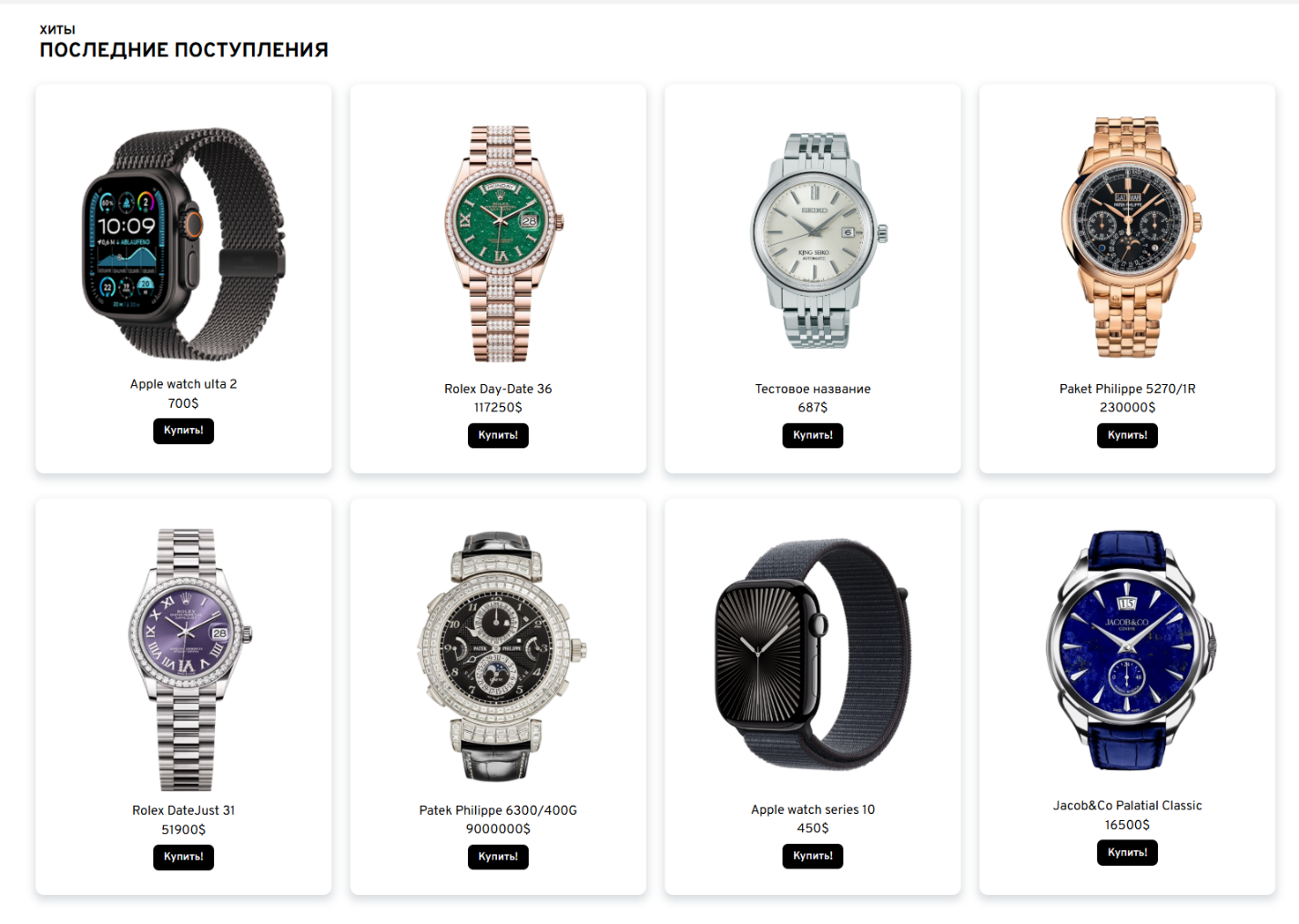


Рисунок 2. Блок, сгенерированный с помощью объектов.

ЗАДАНИЕ 2 (вариант 12).

Реализовать несколько объектов и продемонстрировать 3 основные парадигмы ООП. Реализовать несколько классов и продемонстрировать на них 3 основные парадигмы ООП.

ВЫПОЛНЕНИЕ РАБОТЫ:

Листинг index.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Overpass:ital,wght@0,100..900;1,100..900&display=swap"

        rel="stylesheet">

    <title>Clock Shop</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <header>

        <div class="leftText">

            <h1> ДИЗАЙНЕРСКИЙ ЧАСЫ </h1>

            <p> ПОЛНОСТЬЮ РУЧНОЕ ИЗГОТОВЛЕНИЕ </p>

            <button class="buyHeader buttonToChange openModel" data-price="900000"> КУПИТЬ </button>

            <dialog class="dialogMain">

                <form class="orderForm">

                    <h2>Оформление заказа</h2>

                    <p class="hidden" id="validSuccess"></p>

                    <p class="hidden" id="validError"></p>

                    <p><label for="name">Имя:</label>

                        <input type="text" id="name" name="name" placeholder="Имя" required>

                    <p class="hidden text-danger" id="nameP"></p>

                    </p>

                    <p><label for="secondName">Фамилия:</label>

                        <input type="text" id="secondName" name="secondName" placeholder="Фамилия" required>

                    <p class="hidden text-danger" id="secondNameP"></p>

                    </p>

                    <p><label for="thirdName">Отчество:</label>

                        <input type="text" id="thirdName" name="thirdName" placeholder="Отчество">

                    <p class="hidden text-danger" id="thirdNameP"></p>

                    </p>

                    <p><label for="address">Адрес доставки:</label>

                        <input type="text" id="address" name="address" placeholder="Ул. Пушкино, д. 2, кв. 1" required>

                    </p>

                    <p><label for="quantity">Количество:</label>

                        <input type="number" id="quantity" name="quantity" placeholder="1" min="1" max="10" required>

                    </p>

                    <p><label for="promo">Промокод:</label>

                        <input type="text" id="promo" placeholder="\*Промокод\*" name="promo">

                    </p>

                    <p class="priceHeader"></p>

                    <input type="submit" value="Заказать!">

                </form>

                <button id="closeModel" class="closeModel">&#x2715 </button>

            </dialog>

        </div>

        <div class="rightText">

            <p> JACOB & CO <br>EPIC X TOURBILON BAGUETTE <br>900.000$</p>

            <img src="imgs/mainPhotoWatches.png">

        </div>

    </header>

    <main>

        <div class="textInMain">

            <p>ХИТЫ</p>

            <p>ПОСЛЕДНИЕ ПОСТУПЛЕНИЯ</p>

        </div>

        <div class="productsList"> </div>

        <div class="textInMain" style="margin-top: 2%;">

            <p>ЗАКАЗЫ</p>

        </div>

        <div class="orderContainer">

            <div class="orderHeader">

                <p>Имя</p>

                <p>Фамилия</p>

                <p>Почта</p>

                <p>Адрес</p>

                <p>Количество</p>

            </div>

        </div>

        <!-- <div class="addModule">

            <div class="leftSide">

                <div class="textInMain addText">

                    <p>РАБОТА С ОБЪЕКТАМИ</p>

                </div>

                <form class="decorateBlock" id="objectId">

                    <p><label for="idObjectValue">Id объекта</label>

                        <input type="number" required for="idObjectValue" placeholder="1.." id="idObjectValueInput">

                    </p>

                </form>

                <form class="decorateBlock" id="objectFuncs">

                    <p><label for="priceObjectValue">Цена</label>

                        <input type="number" required for="priceObjectValue" placeholder="100$.."

                            id="priceObjectValueInput">

                    </p>

                    <p><label for="nameObjectValue">Название</label>

                        <input type="text" required for="nameObjectValue" placeholder="Название..."

                            id="nameObjectValueInput">

                    </p>

                </form>

                <form class="decorateBlock" id="objcetButtons">

                    <button id="PriceChanger"> Изменить цену! </button>

                    <button id="NameChanger"> Изменить name! </button>

                </form>

                <p class="objectWorkResult"></p>

            </div>

        </div> -->

    </main>

    <script src="script.js"></script>

</body>

</html>

Листинг script.js:

// Смена цвета кнопки при наведении

const button = document.body.querySelectorAll(".buttonToChange");

for (let i = 0; i < button.length; i++) {

    button[i].addEventListener("mouseover", function () {

        button[i].classList.add("changebackground");

    });

    button[i].addEventListener("mouseout", function () {

        button[i].classList.remove("changebackground");

    });

}

// Модальное окно

const dialog = document.querySelector(".dialogMain");

const html = document.querySelector("html");

const priceHeader = document.querySelector(".priceHeader");

const openModels = document.querySelectorAll(".openModel");

for (let i = 0; i < openModels.length; i++) {

    openModels[i].addEventListener("click", function () {

        dialog.showModal();

        html.classList.add("changehtml");

        priceHeader.textContent = `Цена: ${price.toLocaleString()}$`;

    });

}

document.querySelector("#closeModel").onclick = function () {

    dialog.close();

    html.classList.remove("changehtml");

};

//Изменения цены

const formElement = document.querySelector(".orderForm");

const quantity = document.querySelector("#quantity");

const promo = document.querySelector("#promo");

const promoList = {

    "10procentSaleWOW": 0.9,

    "15yearShop": 0.85,

};

function updatePrice() {

    let quantityVal = 1;

    const price = null;

    if (quantity.value > 1) {

        quantityVal = quantity.value;

    }

    let discount = promo.value in promoList ? promoList[promo.value] : 1;

    priceHeader.textContent = `Цена: ${createPrice(price, quantityVal, discount).toLocaleString()}$ (${quantityVal} шт, скидка ${Math.round((1 - discount) \* 100)}%)`;

}

function createPrice(priceFunc, quantityFunc, discountFunc) {

    let finalPrice = priceFunc \* quantityFunc \* discountFunc;

    return finalPrice;

}

quantity.addEventListener("input", updatePrice);

promo.addEventListener("input", updatePrice);

// Отправка формы

formElement.addEventListener("submit", (event) => {

    event.preventDefault();

    const formData = new FormData(formElement);

    const name = formData.get("name");

    const surname = formData.get("secondName");

    const thirdName = formData.get("thirdName");

    const address = formData.get("address");

    if (!checker(name, "nameP") || !checker(surname, "secondNameP") || !checker(thirdName, "thirdNameP")) {

        document.getElementById("validError").textContent = "Ошибка! Некоторый данный были введены неправильно";

        document.getElementById("validError").classList.remove("hidden");

        document.getElementById("validSuccess").classList.add("hidden");

    } else {

        document.getElementById("validSuccess").textContent = `Спасибо, ${name} ${surname} ${thirdName}!\nВаш заказ на адрес - ${address} в количестве ${quantity.value}шт будет исполнен в ближайшее время!`;

        document.getElementById("validSuccess").classList.remove("hidden");

        document.getElementById("validError").classList.add("hidden");

    }

});

function checker(str, idEl) {

    const regexStr = /^[A-Za-zА-Яа-яЁё]+$/;

    if (!regexStr.test(str)) {

        document.getElementById(idEl).textContent = "Не должно быть цифр!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    if (str.length > 50 || str.length < 3) {

        document.getElementById(idEl).textContent = "От 3 до 50 символов!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    document.getElementById(idEl).textContent = "";

    return true;

}

//Класс продукты

let productList = [];

class Product{

    #id;

    #name;

    #price;

    #imgPath;

    constructor(id, name, price, imgPath){

        this.#id = id;

        this.#name = name;

        this.#price = price;

        this.#imgPath = imgPath;

    }

    get getId(){

        return this.#id;

    }

    get getName(){

        return this.#name;

    }

    get getPrice(){

        return this.#price;

    }

    get getImgPath(){

        return this.#imgPath;

    }

    whatIsIt(){

        console.log("Здесь выводится тип продукта");

    }

}

class Watch extends Product{

    #type = "Часы";

    constructor(id, name, price, imgPath){

        super(id, name, price, imgPath);

    }

    get getType(){

        return this.#type;

    }

    whatIsIt(){

        console.log("Вывелись часы");

    }

}

class Accessory extends Product{

    #type = "Аксессуар";

    constructor(id, name, price, imgPath){

        super(id, name, price, imgPath);

    }

    get getType(){

        return this.#type;

    }

    whatIsIt(){

        console.log("Вывелся аксессуар");

    }

}

productList.push(

    new Watch(1, "Apple watch ulta 2", 700, "imgs/appleulta.png"),

    new Watch(2, "Rolex Day-Date 36", 117250, "imgs/rolexDayDate36.png"),

    new Accessory(3, "118135 RHODIUM", 250, "imgs/RemeshokRolex.png"),

    new Watch(4, "Paket Philippe 5270/1R", 230000, "imgs/PatekPhilippe.png"),

    new Accessory(5, "Mademoiselle Prive H3567", 310 , "imgs/RemeshokChannel.png"),

    new Watch(6, "Patek Philippe 6300/400G", 9000000, "imgs/PatekPhilippeBlackWhite.png"),

    new Watch(7, "Apple watch series 10", 450, "imgs/appleseries10.png"),

    new Watch(8, "Jacob&Co Palatial Classic", 16500, "imgs/Jacob&Co.png")

);

//Класс заказы

let orderList = [];

class Order{

    #name;

    #surname;

    #email;

    #address;

    #quantity;

    constructor(name, surname, email, address, quantity){

        this.#name = name;

        this.#surname = surname;

        this.#email = email;

        this.#address = address;

        this.#quantity = quantity;

    }

    get getName(){

        return this.#name;

    }

    get getSurname(){

        return this.#surname;

    }

    get getEmail(){

        return this.#email;

    }

    get getAddress(){

        return this.#address;

    }

    get getQuantity(){

        return this.#quantity;

    }

}

orderList.push(

    new Order("Аноним", "Анонимов", "Anonim@gmail.com", "Город Анонимов, ул. Анонимная, д. Анон, кв. А", 2),

    new Order("Виктор", "Викторов", "ViktorSupet@mail.ru", "Город Москва, улица Мира, д.4", 9),

    new Order("Тестер", "Тестеров", "tester@top.com", "г.Тест, ул. Тестеров, д. -004", 5)

)

//Генерация html

const productListContainer = document.querySelector(".productsList");

const orderListContainer = document.querySelector(".orderContainer");

productList.forEach((product) => {

    const productUnit = document.createElement("div");

    productUnit.classList.add("productUnit");

    const imgProduct = document.createElement("img");

    imgProduct.src = product.getImgPath;

    const nameProduct = document.createElement("p");

    nameProduct.textContent = product.getName;

    const typeProduct = document.createElement("p");

    typeProduct.textContent = "Тип продукта: " + product.getType;

    const priceProduct = document.createElement("p");

    priceProduct.textContent = `${product.getPrice}$`;

    product.whatIsIt();

    const buyButtonProduct = document.createElement("button");

    buyButtonProduct.classList.add("buttonUnit", "buttonToChange", "openModel");

    buyButtonProduct.textContent = "Купить!";

    buyButtonProduct.dataset.price = product.price;

    buyButtonProduct.addEventListener("click", function () {

        dialog.showModal();

        html.classList.add("changehtml");

        priceHeader.textContent = `Цена: ${product.price.toLocaleString()}$`;

    });

    productUnit.appendChild(imgProduct);

    productUnit.appendChild(nameProduct);

    productUnit.appendChild(typeProduct);

    productUnit.appendChild(priceProduct);

    productUnit.appendChild(buyButtonProduct);

    productListContainer.appendChild(productUnit);

});

orderList.forEach((order) => {

    const orderUnit = document.createElement("div");

    orderUnit.classList.add("orderRow");

    const nameOrder = document.createElement("p");

    nameOrder.textContent = order.getName;

    const surnameOrder = document.createElement("p");

    surnameOrder.textContent = order.getSurname;

    const emailOrder = document.createElement("p");

    emailOrder.textContent = order.getEmail;

    const addressOrder = document.createElement("p");

    addressOrder.textContent = order.getAddress;

    const quantityOrder = document.createElement("p");

    quantityOrder.textContent = order.getQuantity;

    orderUnit.appendChild(nameOrder);

    orderUnit.appendChild(surnameOrder);

    orderUnit.appendChild(emailOrder);

    orderUnit.appendChild(addressOrder);

    orderUnit.appendChild(quantityOrder);

    orderListContainer.appendChild(orderUnit);

});

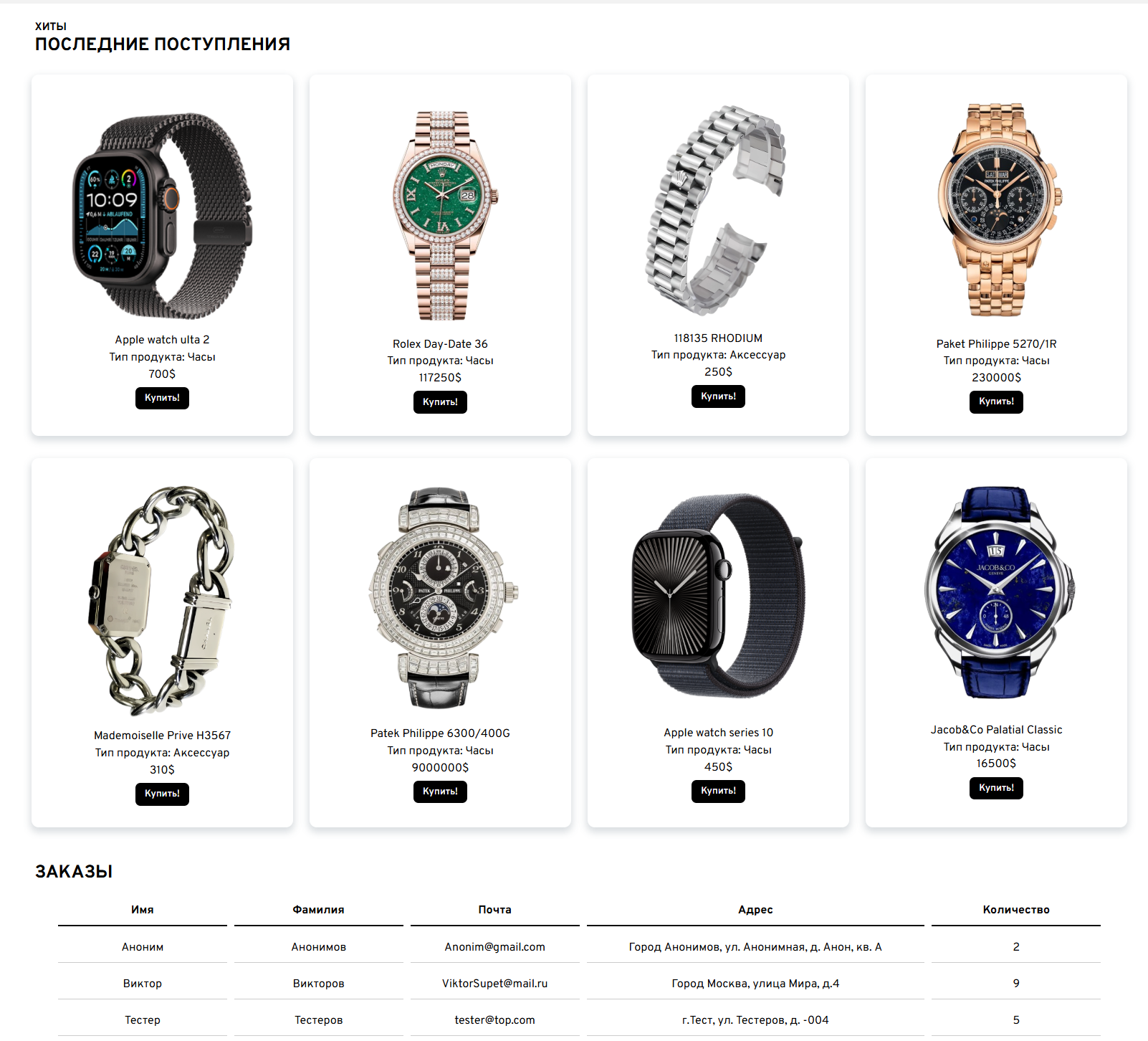


Рисунок 3. Блоки, сгенерированные при помощи классов.

Листинг index.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Overpass:ital,wght@0,100..900;1,100..900&display=swap"

        rel="stylesheet">

    <title>Clock Shop</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <header>

        <div class="leftText">

            <h1> ДИЗАЙНЕРСКИЙ ЧАСЫ </h1>

            <p> ПОЛНОСТЬЮ РУЧНОЕ ИЗГОТОВЛЕНИЕ </p>

            <button class="buyHeader buttonToChange openModel" data-price="900000"> КУПИТЬ </button>

            <dialog class="dialogMain">

                <form class="orderForm">

                    <h2>Оформление заказа</h2>

                    <p class="hidden" id="validSuccess"></p>

                    <p class="hidden" id="validError"></p>

                    <p><label for="name">Имя:</label>

                        <input type="text" id="name" name="name" placeholder="Имя" required>

                    <p class="hidden text-danger" id="nameP"></p>

                    </p>

                    <p><label for="secondName">Фамилия:</label>

                        <input type="text" id="secondName" name="secondName" placeholder="Фамилия" required>

                    <p class="hidden text-danger" id="secondNameP"></p>

                    </p>

                    <p><label for="thirdName">Отчество:</label>

                        <input type="text" id="thirdName" name="thirdName" placeholder="Отчество">

                    <p class="hidden text-danger" id="thirdNameP"></p>

                    </p>

                    <p><label for="address">Адрес доставки:</label>

                        <input type="text" id="address" name="address" placeholder="Ул. Пушкино, д. 2, кв. 1" required>

                    </p>

                    <p><label for="quantity">Количество:</label>

                        <input type="number" id="quantity" name="quantity" placeholder="1" min="1" max="10" required>

                    </p>

                    <p><label for="promo">Промокод:</label>

                        <input type="text" id="promo" placeholder="\*Промокод\*" name="promo">

                    </p>

                    <p class="priceHeader"></p>

                    <input type="submit" value="Заказать!">

                </form>

                <button id="closeModel" class="closeModel">&#x2715 </button>

            </dialog>

        </div>

        <div class="rightText">

            <p> JACOB & CO <br>EPIC X TOURBILON BAGUETTE <br>900.000$</p>

            <img src="imgs/mainPhotoWatches.png">

        </div>

    </header>

    <main>

        <div class="textInMain">

            <p>ХИТЫ</p>

            <p>ПОСЛЕДНИЕ ПОСТУПЛЕНИЯ</p>

        </div>

        <div class="productsList"> </div>

        <div class="textInMain" style="margin-top: 2%;">

            <p>ЗАКАЗЫ</p>

        </div>

        <div class="orderContainer">

            <div class="orderHeader">

                <p>Имя</p>

                <p>Фамилия</p>

                <p>Почта</p>

                <p>Адрес</p>

                <p>Количество</p>

            </div>

        </div>

        <!-- <div class="addModule">

            <div class="leftSide">

                <div class="textInMain addText">

                    <p>РАБОТА С ОБЪЕКТАМИ</p>

                </div>

                <form class="decorateBlock" id="objectId">

                    <p><label for="idObjectValue">Id объекта</label>

                        <input type="number" required for="idObjectValue" placeholder="1.." id="idObjectValueInput">

                    </p>

                </form>

                <form class="decorateBlock" id="objectFuncs">

                    <p><label for="priceObjectValue">Цена</label>

                        <input type="number" required for="priceObjectValue" placeholder="100$.."

                            id="priceObjectValueInput">

                    </p>

                    <p><label for="nameObjectValue">Название</label>

                        <input type="text" required for="nameObjectValue" placeholder="Название..."

                            id="nameObjectValueInput">

                    </p>

                </form>

                <form class="decorateBlock" id="objcetButtons">

                    <button id="PriceChanger"> Изменить цену! </button>

                    <button id="NameChanger"> Изменить name! </button>

                </form>

                <p class="objectWorkResult"></p>

            </div>

        </div> -->

    </main>

    <script src="script.js"></script>

</body>

</html>

Листинг script.js:

// Смена цвета кнопки при наведении

const button = document.body.querySelectorAll(".buttonToChange");

for (let i = 0; i < button.length; i++) {

    button[i].addEventListener("mouseover", function () {

        button[i].classList.add("changebackground");

    });

    button[i].addEventListener("mouseout", function () {

        button[i].classList.remove("changebackground");

    });

}

// Модальное окно

const dialog = document.querySelector(".dialogMain");

const html = document.querySelector("html");

const priceHeader = document.querySelector(".priceHeader");

const openModels = document.querySelectorAll(".openModel");

for (let i = 0; i < openModels.length; i++) {

    openModels[i].addEventListener("click", function () {

        dialog.showModal();

        html.classList.add("changehtml");

        priceHeader.textContent = `Цена: ${price.toLocaleString()}$`;

    });

}

document.querySelector("#closeModel").onclick = function () {

    dialog.close();

    html.classList.remove("changehtml");

};

//Изменения цены

const formElement = document.querySelector(".orderForm");

const quantity = document.querySelector("#quantity");

const promo = document.querySelector("#promo");

const promoList = {

    "10procentSaleWOW": 0.9,

    "15yearShop": 0.85,

};

function updatePrice() {

    let quantityVal = 1;

    const price = null;

    if (quantity.value > 1) {

        quantityVal = quantity.value;

    }

    let discount = promo.value in promoList ? promoList[promo.value] : 1;

    priceHeader.textContent = `Цена: ${createPrice(price, quantityVal, discount).toLocaleString()}$ (${quantityVal} шт, скидка ${Math.round((1 - discount) \* 100)}%)`;

}

function createPrice(priceFunc, quantityFunc, discountFunc) {

    let finalPrice = priceFunc \* quantityFunc \* discountFunc;

    return finalPrice;

}

quantity.addEventListener("input", updatePrice);

promo.addEventListener("input", updatePrice);

// Отправка формы

formElement.addEventListener("submit", (event) => {

    event.preventDefault();

    const formData = new FormData(formElement);

    const name = formData.get("name");

    const surname = formData.get("secondName");

    const thirdName = formData.get("thirdName");

    const address = formData.get("address");

    if (!checker(name, "nameP") || !checker(surname, "secondNameP") || !checker(thirdName, "thirdNameP")) {

        document.getElementById("validError").textContent = "Ошибка! Некоторый данный были введены неправильно";

        document.getElementById("validError").classList.remove("hidden");

        document.getElementById("validSuccess").classList.add("hidden");

    } else {

        document.getElementById("validSuccess").textContent = `Спасибо, ${name} ${surname} ${thirdName}!\nВаш заказ на адрес - ${address} в количестве ${quantity.value}шт будет исполнен в ближайшее время!`;

        document.getElementById("validSuccess").classList.remove("hidden");

        document.getElementById("validError").classList.add("hidden");

    }

});

function checker(str, idEl) {

    const regexStr = /^[A-Za-zА-Яа-яЁё]+$/;

    if (!regexStr.test(str)) {

        document.getElementById(idEl).textContent = "Не должно быть цифр!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    if (str.length > 50 || str.length < 3) {

        document.getElementById(idEl).textContent = "От 3 до 50 символов!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    document.getElementById(idEl).textContent = "";

    return true;

}

//Объект продукты

let productList = [];

function Product(id, name, price, imgPath) {

    this.id = id;

    this.name = name;

    this.price = price;

    this.imgPath = imgPath;

    this.getName = function () {

        return this.name;

    };

    this.getPrice = function () {

        return this.price;

    };

    this.getImgPath = function () {

        return this.imgPath;

    };

    this.getId = function () {

        return this.id;

    };

}

function Watch(id, name, price, imgPath) {

    Product.call(this, id, name, price, imgPath);

    this.type = "Часы";

    this.getType = function () {

        return this.type;

    };

    this.whatIsIt = function () {

        console.log("Вывелись часы");

    };

}

Watch.prototype = Object.create(Product.prototype);

Watch.prototype.constructor = Product;

function Accessory(id, name, price, imgPath) {

    Product.call(this, id, name, price, imgPath);

    this.type = "Аксессуар";

    this.getType = function () {

        return this.type;

    };

    this.whatIsIt = function () {

        console.log("Вывелся аксессуар");

    };

}

Accessory.prototype = Object.create(Product.prototype);

Accessory.prototype.constructor = Product;

productList.push(

    new Watch(1, "Apple watch ulta 2", 700, "imgs/appleulta.png"),

    new Watch(2, "Rolex Day-Date 36", 117250, "imgs/rolexDayDate36.png"),

    new Accessory(3, "118135 RHODIUM", 250, "imgs/RemeshokRolex.png"),

    new Watch(4, "Paket Philippe 5270/1R", 230000, "imgs/PatekPhilippe.png"),

    new Accessory(5, "Mademoiselle Prive H3567", 310, "imgs/RemeshokChannel.png"),

    new Watch(6, "Patek Philippe 6300/400G", 9000000, "imgs/PatekPhilippeBlackWhite.png"),

    new Watch(7, "Apple watch series 10", 450, "imgs/appleseries10.png"),

    new Watch(8, "Jacob&Co Palatial Classic", 16500, "imgs/Jacob&Co.png")

);

//Объект заказы

let orderList = [];

function Order(name, surname, email, address, quantity) {

    let nameOrder = name;

    let surnameOrder = surname;

    let emailOrder = email;

    let addressOrder = address;

    let quantityOrder = quantity;

    this.getName = function () {

        return nameOrder;

    };

    this.getSurname = function () {

        return surnameOrder;

    };

    this.getEmail = function () {

        return emailOrder;

    };

    this.getAddress = function () {

        return addressOrder;

    };

    this.getQuantity = function () {

        return quantityOrder;

    };

}

orderList.push(

    new Order("Аноним", "Анонимов", "Anonim@gmail.com", "Город Анонимов, ул. Анонимная, д. Анон, кв. А", 2),

    new Order("Виктор", "Викторов", "ViktorSupet@mail.ru", "Город Москва, улица Мира, д.4", 9),

    new Order("Тестер", "Тестеров", "tester@top.com", "г.Тест, ул. Тестеров, д. -004", 5)

);

//Генерация html

const productListContainer = document.querySelector(".productsList");

const orderListContainer = document.querySelector(".orderContainer");

productList.forEach((product) => {

    const productUnit = document.createElement("div");

    productUnit.classList.add("productUnit");

    const imgProduct = document.createElement("img");

    imgProduct.src = product.getImgPath();

    const nameProduct = document.createElement("p");

    nameProduct.textContent = product.getName();

    const typeProduct = document.createElement("p");

    typeProduct.textContent = "Тип продукта: " + product.getType();

    const priceProduct = document.createElement("p");

    priceProduct.textContent = `${product.getPrice()}$`;

    product.whatIsIt();

    const buyButtonProduct = document.createElement("button");

    buyButtonProduct.classList.add("buttonUnit", "buttonToChange", "openModel");

    buyButtonProduct.textContent = "Купить!";

    buyButtonProduct.dataset.price = product.price;

    buyButtonProduct.addEventListener("click", function () {

        dialog.showModal();

        html.classList.add("changehtml");

        priceHeader.textContent = `Цена: ${product.price.toLocaleString()}$`;

    });

    productUnit.appendChild(imgProduct);

    productUnit.appendChild(nameProduct);

    productUnit.appendChild(typeProduct);

    productUnit.appendChild(priceProduct);

    productUnit.appendChild(buyButtonProduct);

    productListContainer.appendChild(productUnit);

});

orderList.forEach((order) => {

    const orderUnit = document.createElement("div");

    orderUnit.classList.add("orderRow");

    const nameOrder = document.createElement("p");

    nameOrder.textContent = order.getName();

    const surnameOrder = document.createElement("p");

    surnameOrder.textContent = order.getSurname();

    const emailOrder = document.createElement("p");

    emailOrder.textContent = order.getEmail();

    const addressOrder = document.createElement("p");

    addressOrder.textContent = order.getAddress();

    const quantityOrder = document.createElement("p");

    quantityOrder.textContent = order.getQuantity();

    orderUnit.appendChild(nameOrder);

    orderUnit.appendChild(surnameOrder);

    orderUnit.appendChild(emailOrder);

    orderUnit.appendChild(addressOrder);

    orderUnit.appendChild(quantityOrder);

    orderListContainer.appendChild(orderUnit);

});

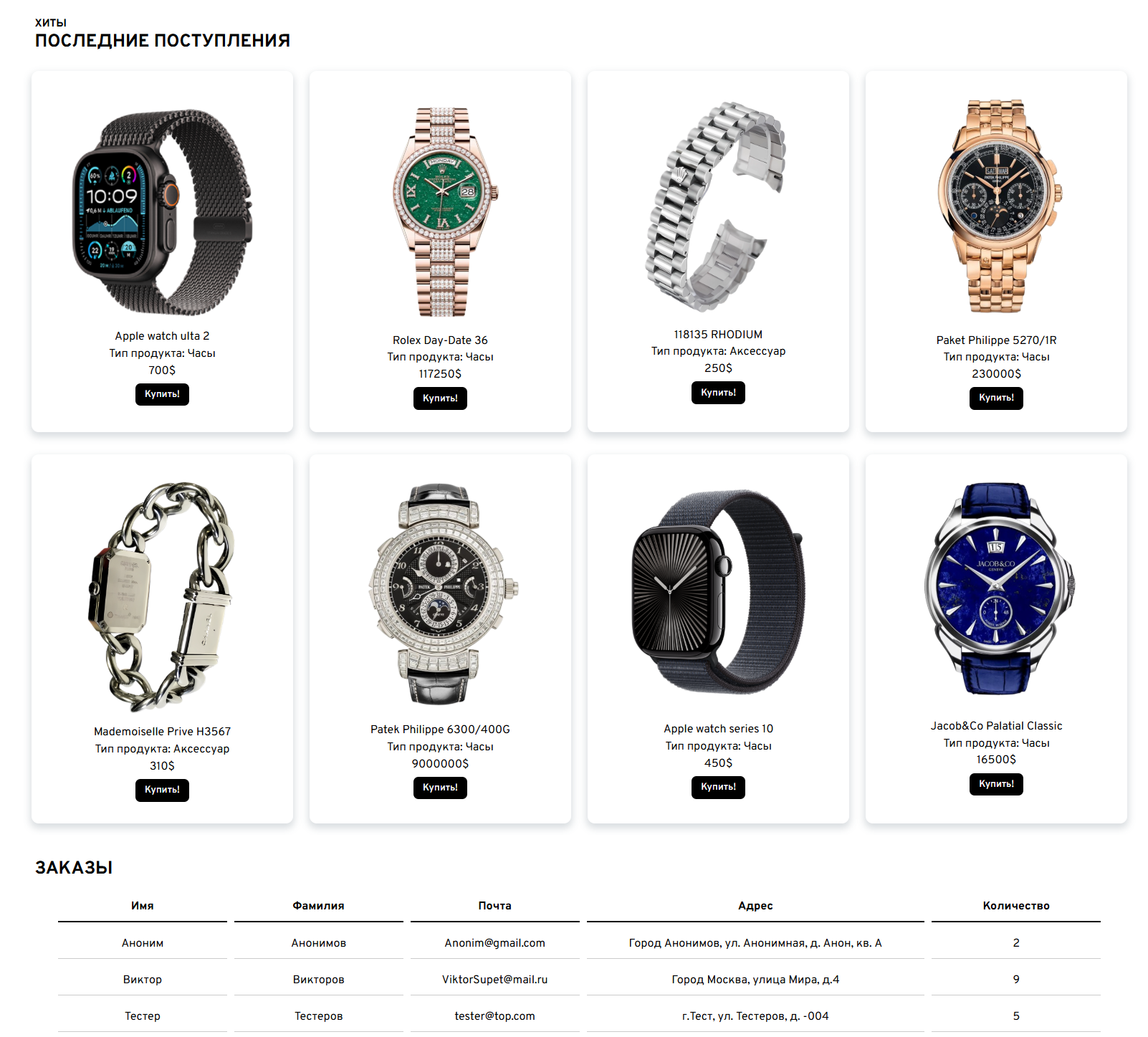


Рисунок 4. Блоки, сгенерированные при помощи объектов.

ЗАДАНИЕ 3 (вариант 12).

Обработать исключения с помощью блока «try…catch», генерации собственных ошибок, «try…catch…finally».

ВЫПОЛНЕНИЕ РАБОТЫ:

Листинг index.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Overpass:ital,wght@0,100..900;1,100..900&display=swap"

        rel="stylesheet">

    <title>Clock Shop</title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <header>

        <div class="leftText">

            <h1> ДИЗАЙНЕРСКИЙ ЧАСЫ </h1>

            <p> ПОЛНОСТЬЮ РУЧНОЕ ИЗГОТОВЛЕНИЕ </p>

            <button class="buyHeader buttonToChange openModel" data-price="900000"> КУПИТЬ </button>

            <dialog class="dialogMain">

                <form class="orderForm">

                    <h2>Оформление заказа</h2>

                    <p class="hidden" id="validSuccess"></p>

                    <p class="hidden" id="validError"></p>

                    <p><label for="name">Имя:</label>

                        <input type="text" id="name" name="name" placeholder="Имя" required>

                    <p class="hidden text-danger" id="nameP"></p>

                    </p>

                    <p><label for="secondName">Фамилия:</label>

                        <input type="text" id="secondName" name="secondName" placeholder="Фамилия" required>

                    <p class="hidden text-danger" id="secondNameP"></p>

                    </p>

                    <p><label for="thirdName">Отчество:</label>

                        <input type="text" id="thirdName" name="thirdName" placeholder="Отчество">

                    <p class="hidden text-danger" id="thirdNameP"></p>

                    </p>

                    <p><label for="address">Адрес доставки:</label>

                        <input type="text" id="address" name="address" placeholder="Ул. Пушкино, д. 2, кв. 1" required>

                    </p>

                    <p><label for="quantity">Количество:</label>

                        <input type="number" id="quantity" name="quantity" placeholder="1" min="1" max="10" required>

                    </p>

                    <p><label for="promo">Промокод:</label>

                        <input type="text" id="promo" placeholder="\*Промокод\*" name="promo">

                    </p>

                    <p class="priceHeader"></p>

                    <input type="submit" value="Заказать!">

                </form>

                <button id="closeModel" class="closeModel">&#x2715 </button>

            </dialog>

        </div>

        <div class="rightText">

            <p> JACOB & CO <br>EPIC X TOURBILON BAGUETTE <br>900.000$</p>

            <img src="imgs/mainPhotoWatches.png">

        </div>

    </header>

    <main>

        <div class="textInMain">

            <p>ХИТЫ</p>

            <p id="ProductCount">ПОСЛЕДНИЕ ПОСТУПЛЕНИЯ</p>

        </div>

        <div class="productsList"> </div>

        <div class="textInMain" style="margin-top: 2%;">

            <p>ЗАКАЗЫ</p>

        </div>

        <div class="orderContainer">

            <div class="orderHeader">

                <p>Имя</p>

                <p>Фамилия</p>

                <p>Почта</p>

                <p>Адрес</p>

                <p>Количество</p>

            </div>

        </div>

        <div class="addModule">

            <div class="leftSide">

                <div class="textInMain addText">

                    <p>РАБОТА С ФАЙЛОМ</p>

                </div>

                <form class="decorateBlock" id="fileWork">

                    <p><label for="fileGet">Файл</label>

                        <input type="file" required for="fileGet" id="fileGetInput">

                    </p>

                    <button id="fileRead"> Прочитать! </button>

                </form>

                <p id="fileWorkResult" class="resultDecorate"></p>

            </div>

        </div>

    </main>

    <script src="script.js"></script>

</body>

</html>

Листинг script.js:

"use strict";

// Смена цвета кнопки при наведении

const button = document.body.querySelectorAll(".buttonToChange");

for (let i = 0; i < button.length; i++) {

    button[i].addEventListener("mouseover", function () {

        button[i].classList.add("changebackground");

    });

    button[i].addEventListener("mouseout", function () {

        button[i].classList.remove("changebackground");

    });

}

// Модальное окно

const dialog = document.querySelector(".dialogMain");

const html = document.querySelector("html");

const priceHeader = document.querySelector(".priceHeader");

const openModels = document.querySelectorAll(".openModel");

for (let i = 0; i < openModels.length; i++) {

    openModels[i].addEventListener("click", function () {

        dialog.showModal();

        html.classList.add("changehtml");

        priceHeader.textContent = `Цена: ${price.toLocaleString()}$`;

    });

}

document.querySelector("#closeModel").onclick = function () {

    dialog.close();

    html.classList.remove("changehtml");

};

//Изменения цены

const formElement = document.querySelector(".orderForm");

const quantity = document.querySelector("#quantity");

const promo = document.querySelector("#promo");

const promoList = {

    "10procentSaleWOW": 0.9,

    "15yearShop": 0.85,

};

function updatePrice() {

    let quantityVal = 1;

    const price = null;

    if (quantity.value > 1) {

        quantityVal = quantity.value;

    }

    let discount = promo.value in promoList ? promoList[promo.value] : 1;

    priceHeader.textContent = `Цена: ${createPrice(price, quantityVal, discount).toLocaleString()}$ (${quantityVal} шт, скидка ${Math.round((1 - discount) \* 100)}%)`;

}

function createPrice(priceFunc, quantityFunc, discountFunc) {

    let finalPrice = priceFunc \* quantityFunc \* discountFunc;

    return finalPrice;

}

quantity.addEventListener("input", updatePrice);

promo.addEventListener("input", updatePrice);

// Отправка формы

formElement.addEventListener("submit", (event) => {

    event.preventDefault();

    const formData = new FormData(formElement);

    const name = formData.get("name");

    const surname = formData.get("secondName");

    const thirdName = formData.get("thirdName");

    const address = formData.get("address");

    if (!checker(name, "nameP") || !checker(surname, "secondNameP") || !checker(thirdName, "thirdNameP")) {

        document.getElementById("validError").textContent = "Ошибка! Некоторый данный были введены неправильно";

        document.getElementById("validError").classList.remove("hidden");

        document.getElementById("validSuccess").classList.add("hidden");

    } else {

        document.getElementById("validSuccess").textContent = `Спасибо, ${name} ${surname} ${thirdName}!\nВаш заказ на адрес - ${address} в количестве ${quantity.value}шт будет исполнен в ближайшее время!`;

        document.getElementById("validSuccess").classList.remove("hidden");

        document.getElementById("validError").classList.add("hidden");

    }

});

function checker(str, idEl) {

    const regexStr = /^[A-Za-zА-Яа-яЁё]+$/;

    if (!regexStr.test(str)) {

        document.getElementById(idEl).textContent = "Не должно быть цифр!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    if (str.length > 50 || str.length < 3) {

        document.getElementById(idEl).textContent = "От 3 до 50 символов!";

        document.getElementById(idEl).classList.remove("hidden");

        return false;

    }

    document.getElementById(idEl).textContent = "";

    return true;

}

//Класс продукты

let productList = [];

class Product {

    #id;

    #name;

    #price;

    #imgPath;

    constructor(id, name, price, imgPath) {

        this.#id = id;

        this.#name = name;

        this.#price = price;

        this.#imgPath = imgPath;

    }

    get getId() {

        return this.#id;

    }

    get getName() {

        return this.#name;

    }

    get getPrice() {

        return this.#price;

    }

    get getImgPath() {

        return this.#imgPath;

    }

    whatIsIt() {

        console.log("Здесь выводится тип продукта");

    }

}

class Watch extends Product {

    #type = "Часы";

    constructor(id, name, price, imgPath) {

        super(id, name, price, imgPath);

    }

    get getType() {

        return this.#type;

    }

    whatIsIt() {

        console.log("Вывелись часы");

    }

}

class Accessory extends Product {

    #type = "Аксессуар";

    constructor(id, name, price, imgPath) {

        super(id, name, price, imgPath);

    }

    get getType() {

        return this.#type;

    }

    whatIsIt() {

        console.log("Вывелся аксессуар");

    }

}

productList.push(

    new Watch(1, "Apple watch ulta 2", 700, "imgs/appleulta.png"),

    new Watch(2, "Rolex Day-Date 36", 117250, "imgs/rolexDayDate36.png"),

    new Accessory(3, "118135 RHODIUM", 250, "imgs/RemeshokRolex.png"),

    new Watch(4, "Paket Philippe 5270/1R", 230000, "imgs/PatekPhilippe.png"),

    new Accessory(5, "Mademoiselle Prive H3567", 310, "imgs/RemeshokChannel.png"),

    new Watch(6, "Patek Philippe 6300/400G", 9000000, "imgs/PatekPhilippeBlackWhite.png"),

    new Watch(7, "Apple watch series 10", 450, "imgs/appleseries10.png"),

    new Watch(8, "Jacob&Co Palatial Classic", 16500, "imgs/Jacob&Co.png")

);

//Класс заказы

let orderList = [];

class Order {

    #name;

    #surname;

    #email;

    #address;

    #quantity;

    constructor(name, surname, email, address, quantity) {

        this.#name = name;

        this.#surname = surname;

        this.#email = email;

        this.#address = address;

        this.#quantity = quantity;

    }

    get getName() {

        return this.#name;

    }

    get getSurname() {

        return this.#surname;

    }

    get getEmail() {

        return this.#email;

    }

    get getAddress() {

        return this.#address;

    }

    get getQuantity() {

        return this.#quantity;

    }

}

orderList.push(

    new Order("Аноним", "Анонимов", "Anonim@gmail.com", "Город Анонимов, ул. Анонимная, д. Анон, кв. А", 2),

    new Order("Виктор", "Викторов", "ViktorSupet@mail.ru", "Город Москва, улица Мира, д.4", 9),

    new Order("Тестер", "Тестеров", "tester@top.com", "г.Тест, ул. Тестеров, д. -004", 5)

);

//Генерация html

const productListContainer = document.querySelector(".productsList");

const orderListContainer = document.querySelector(".orderContainer");

let countProducts = 0;

try{

    productList.forEach((product) => {

        const productUnit = document.createElement("div");

        productUnit.classList.add("productUnit");

        const imgProduct = document.createElement("img");

        imgProduct.src = product.getImgPath;

        const nameProduct = document.createElement("p");

        nameProduct.textContent = product.getName;

        const typeProduct = document.createElement("p");

        typeProduct.textContent = "Тип продукта: " + product.getType;

        const priceProduct = document.createElement("p");

        priceProduct.textContent = `${product.getPrice}$`;

        product.whatIsIt();

        const buyButtonProduct = document.createElement("button");

        buyButtonProduct.classList.add("buttonUnit", "buttonToChange", "openModel");

        buyButtonProduct.textContent = "Купить!";

        buyButtonProduct.dataset.price = product.price;

        buyButtonProduct.addEventListener("click", function () {

            dialog.showModal();

            html.classList.add("changehtml");

            priceHeader.textContent = `Цена: ${product.price.toLocaleString()}$`;

        });

        productUnit.appendChild(imgProduct);

        productUnit.appendChild(nameProduct);

        productUnit.appendChild(typeProduct);

        productUnit.appendChild(priceProduct);

        productUnit.appendChild(buyButtonProduct);

        productListContainer.appendChild(productUnit);

        countProducts++;

    });

}

catch (error){

    console.log("Ошибка при выводе товаров!");

}

finally {

    document.getElementById("ProductCount").textContent += ` (${countProducts} шт.)`;

}

orderList.forEach((order) => {

    const orderUnit = document.createElement("div");

    orderUnit.classList.add("orderRow");

    const nameOrder = document.createElement("p");

    nameOrder.textContent = order.getName;

    const surnameOrder = document.createElement("p");

    surnameOrder.textContent = order.getSurname;

    const emailOrder = document.createElement("p");

    emailOrder.textContent = order.getEmail;

    const addressOrder = document.createElement("p");

    addressOrder.textContent = order.getAddress;

    const quantityOrder = document.createElement("p");

    quantityOrder.textContent = order.getQuantity;

    orderUnit.appendChild(nameOrder);

    orderUnit.appendChild(surnameOrder);

    orderUnit.appendChild(emailOrder);

    orderUnit.appendChild(addressOrder);

    orderUnit.appendChild(quantityOrder);

    orderListContainer.appendChild(orderUnit);

});

//Работа с файлом

const fileReadButton = document.getElementById("fileRead");

fileReadButton.addEventListener("click", function(event){

    event.preventDefault();

    document.getElementById("fileWorkResult").textContent = "";

    try{

        const fileTemp = document.getElementById("fileGetInput");

        if (!fileTemp.files.length){

            throw new Error("Файл не выбран!");

        }

        const file = fileTemp.files[0];

        if (file.type != "text/plain"){

            throw new Error("Выбран формат не .txt");

        }

        const reader = new FileReader();

        reader.onload = function(e) {

            document.getElementById("fileWorkResult").textContent = e.target.result;

        };

        reader.readAsText(file);

    }

    catch (error) {

        document.getElementById("fileWorkResult").textContent = error.message;

    }

});

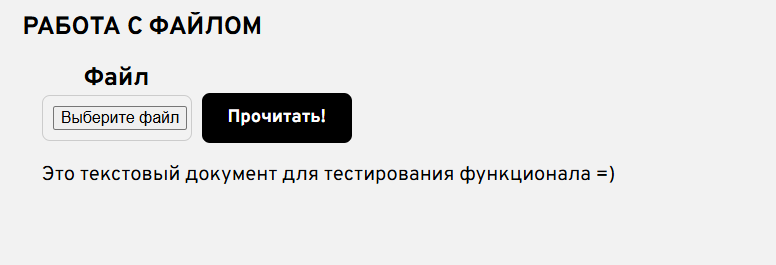


Рисунок 5. Обработка ошибок. Успешное выполнение.

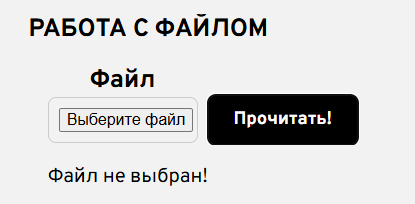


Рисунок 6. Обработка ошибок. Ошибка, если файл не выбран.

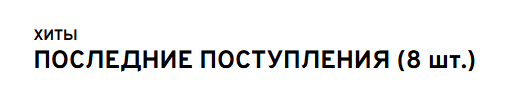


Рисунок 7. Исполнение блока finally.

ВЫВОД К РАБОТЕ:

Я научился решать задачи с помощью объектов и создавать конструкторы объектов с набором общих методом. Решил задачи на ООП, написал код с использованием классов. Выполнил задачи на обработку исключений.